Real-time Multi-Camera Analytics for Traffic Information Extraction and Visualization

Mahshid Ghasemi†, Sofia Kleisarchaki‡, Thomas Calmant‡, Jiawei Lu†,
Shivam Ojha†, Zoran Kostic†, Levent Gürgen‡, Gil Zussman†, Javad Ghaderi†
†Electrical Engineering, Columbia University, ‡Kentyou, Grenoble, France
{mahshid.ghasemi,jl5999,so2639,zk2172,gil.zussman,jghaderi}@columbia.edu
,{sofia.kleisarchaki,thomas.calmant,levent}@kentyou.com

***Abstract*—The density and complexity of urban environments present significant challenges for autonomous vehicles. Moreover, ensuring pedestrians' safety and protecting personal privacy are crucial considerations in these environments. Smart city intersections and AI-powered traffic management systems will be essential for addressing these challenges. Therefore, our research focuses on creating an experimental framework for the design of applications that support the secure and efficient management of traffic intersections in urban areas. We integrated two cameras (street-level and bird's eye view), both viewing an intersection, and a programmable edge computing node, deployed within the COSMOS testbed in New York City, with a central management platform provided by Kentyou. We designed a pipeline to collect and analyze the video streams from both cameras and obtain real-time traffic/pedestrian-related information to support smart city applications. The obtained information from both cameras is merged, and the results are sent to a dedicated dashboard for real-time visualization and further assessment (e.g., accident prevention). The process does not require sending the raw videos in order to avoid violating pedestrians' privacy. In this demo, we present the designed video analytic pipelines and their integration with Kentyou central management platform.**

***Index Terms*—object detection and tracking, camera networks, smart intersection, real-time visualization**

I. Introduction

This demonstration aims to examine the potential of utilizing smart-city technology to develop safe and efficient intersections while also preserving road users' privacy. We implemented a video analytics pipeline (shown in Fig. 1) that processes the real-time video streams of traffic cameras, using AI algorithms, to identify and detect pedestrians/vehicles, the directions, and speed of their movement, traffic lights status, etc., in real-time. Specifically, we use two of the NSF PAWR COSMOS testbed cameras [1]–[3], deployed at the 2^{nd} and 12^{th} floor of Columbia's Mudd building (Fig. 2) looking at the intersection of 120^{th} St. and Amsterdam Ave. in New York City. The real-time stream from the cameras is transmitted to two edge servers for processing, with one server handling the data from each camera on its local network. The edge servers continuously extract traffic/pedestrian-related information from a batch of frames from the corresponding cameras. One of the servers integrates the obtained information from both cameras and creates an MQTT [4] message for each batch of frames. Each MQTT message includes the coordinates, IDs, speed, and direction of all detected pedestrians/vehicles and the status of traffic lights. Only the MQTT messages are sent to the central management platform, Eclipse sensiNact [5], developed by Kentyou, and therefore, privacy is preserved. The platform can use this data and other data sources (e.g., weather stations) for traffic anomaly detection, traffic-light cycle management, accident detection, etc. It also visualizes the real-time traffic/pedestrian-related information using Kentyou User Interface (UI) (Fig. 3) and the historical information using Chronograf UI (Fig. 4). This demo extends our previous work [6] to a multi-camera system and provides significantly improved visualization as well as accident/anomaly prevention.

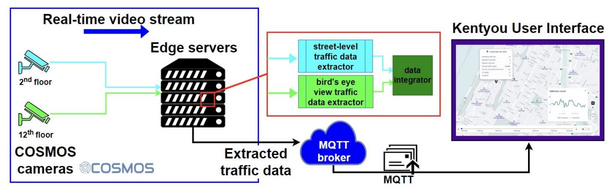

Fig. 1: Video analytics and visualization pipeline: The video streams are analyzed in edge servers, and then the collected data is sent to Kentyou central management platform.

Fig. 2: COSMOS cameras deployed on the 2^{nd} and 12^{th} floor of Columbia's Mudd building and their view of the intersection.

There have been several efforts to use video analytics for traffic surveillance in dense urban areas, including automated detection of helmets on motorcyclists [7], detection of accidents/collisions in traffic videos [8], and alerts following anomalous traffic patterns [9]. A survey on visual traffic surveillance systems appears in [10]. In this demo, we improve upon state-of-the-art by providing a comprehensive view of the intersection. This is achieved by aggregating traffic surveillance detection results, weather information, and other analytics to better deal with safety concerns. Moreover, *the demo demonstrates the ability of the COSMOS platform to share several real-time anonymized information streams with external systems and the ability of Kentyou's platform to interface with various external sensors*.

Below, we briefly describe the analytics components of the video analytic pipeline (Section II), the communication between COSMOS and Kentyou management platform (Section III), and the demonstration details (Section IV).

II. Traffic/Pedestrians Data Analytic Pipeline

In this section, we briefly describe the video analytic pipelines used to extract and integrate the traffic/pedestrian-related information from the 2^{nd} floor (street-level) and 12^{th} floor (bird's eye view) cameras. Due to the significant difference between the bird's eye and street-level view characteristics, different video analytic pipelines are required to obtain the traffic/pedestrian-related information from the two cameras.

A. Street-level View Video Analysis

The street-level view video analytic pipeline, deployed in an edge server as part of the COSMOS network, includes multiple modules:

- **Object detection and tracking:** detects and tracks pedestrians/vehicles. It uses YOLOv4 [11] object detector model and Nvidia DCF-based tracker.
- **Calibration**: uses multi-area calibration method [12] to convert on-image pixel coordinates to on-ground coordinates.
- **Traffic data extractor:** analyzes the output of the object detector and tracking module and derives traffic/pedestrian-related information such as speed and direction.

B. Bird's Eye View Video Analysis

The bird's eye view video analytic pipeline, which is deployed in an edge server also consists of multiple modules:

- **Bird's eye view calibration:** transforms distorted traffic intersection scene into rectangular one perpendicular to the ground with uniform scale. On-ground distances are obtained from on-image distances by a constant scaling factor.
- **Object detection and tracking:** a custom-trained YOLOv4 model on a data set recorded from the same camera [13] along with the NVIDIA DCF-based tracker is used for object detection and tracking on the calibrated video frames.
- **Traffic data extractor:** traffic/pedestrian-related information, such as the position and velocity of pedestrians and vehicles, are computed from the data obtained by the object detection and tracking module.

C. Traffic Light Status Detection

Research on real-time traffic light status detection has practical significance and potential for improving road traffic safety. We use the real-time video stream from the 2^{nd} floor camera for traffic light status detection. The varying luminosity of the traffic lights and their relative distance from the camera

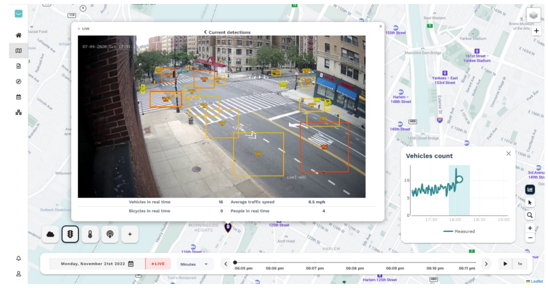

Fig. 3: A sample UI illustrating the data collected and analyzed from the COSMOS testbed.

Fig. 4: Dashboard providing the statistical information about the intersection.

poses some challenges. For example, as highlighted in Fig. 5, there is a lot of glare around the traffic lights at night, which can obstruct the view of the entire traffic light.

Since the camera is static, the location of traffic lights can be found offline manually. After locating the traffic lights, their status is detected as follows:

- **Apply Top-Hat filter:** Top-Hat filter (from OpenCV [14]) is applied to the gray-scale frame, thereby helping to mitigate the image's uneven illumination.
- **Apply binary threshold:** a binary threshold [14] is applied to convert all pixels to either black or white.
- **Retrieve the bright pixel percentage rates:** bright pixel percentage rates are obtained for each traffic light color by counting the total number of white pixels and dividing by the total number of pixels in that traffic light color frame. The light with the highest rate is considered to be status.

The traffic light detection algorithm achieves an accuracy of 99.58% on a validation set consisting of videos from various times of day, with the ground truth determined through manual labeling of the video streams.

D. Integration of Street-level and Bird's Eye View

The calculation of on-ground distance is relatively simple when using a bird's eye view compared to a street-level perspective. In addition, object occlusion is infrequent in a bird's eye view, resulting in an unobstructed view of objects. However, pedestrian detection is more accurate from a street-level perspective because pedestrians appear smaller from a bird's eye view. To address this issue, we integrate the results obtained from both cameras to provide more precise traffic and pedestrian-related information compared to using only a single view. Below, we describe the steps for this integration:

- **Temporal alignment:** find the frames with similar timestamps from the two cameras (to support integration).
- **Spatial alignment:** accurate on-ground coordinate calculation from the 2^{nd} floor camera requires multi-area calibration [12], i.e., calibrating each area individually. Therefore, to achieve temporal alignment, we determined the

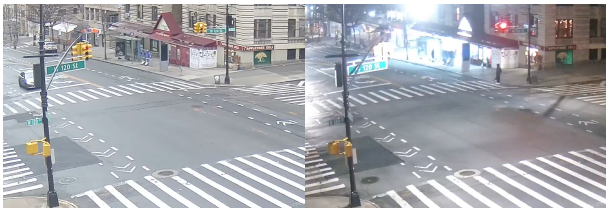

Fig. 5: Video stream frames (day and night), highlighting the challenges related to traffic light status detection.

Fig. 6: Detection areas of 2nd floor and 12th floor cameras.

Fig. 7: Monitored areas (in orange) for accident prevention.

corresponding areas from the perspective of the 12th floor camera as well (illustrated in Fig. 6). For each detected pedestrian/vehicle on both cameras, the area it lies in, and its on-ground coordinates with respect to the origin of that area are calculated. This allows objects detected in different views to be spatially aligned.

- **Integration:** all the pedestrians/vehicles detected from both cameras are integrated, and the objects with high occlusion are removed.

The obtained information is sent to Kentyou central management platform for statistical analysis and visualization.

III. Real-time Information Visualization and Accident Prevention

Kentyou management platform provides a real-time representation of the intersection (Fig. 3) without revealing private information. The visualization depicts real-time traffic information, including the current number of vehicles, bicycles, and pedestrians, average traffic speed, accidents that occurred this month, weather conditions (temperature, clouds, etc.), and traffic violations (speed and red light). Moreover, it provides a dedicated visualization for accident prevention.

The accident prevention mechanism employs simultaneous monitoring and analysis of data from three distinct areas of the intersection. Each such area A, shown in Fig. 7 as orange rectangle, is monitored for pedestrian P. If detected, the surrounding area of A is monitored until P exits safely. If a vehicle is detected in the surrounding area during monitoring, its distance from the pedestrian is calculated. If this distance is below a threshold r, a warning is raised, and the corresponding rectangle is colored red in Kentyou UI.

The prevention mechanism is adjusted for high sensitivity (true positive rate) by properly tuning A and r. Evaluation on synthetic data achieved 100% sensitivity, while evaluation on real data raised 1.2 false alarms per hour. The results indicate a satisfactory balance between missing potential accidents and excessive alerts.

IV. Demonstration

The demonstration showcases the design and performance of the end-to-end pipeline, starting with data collection at the COSMOS testbed and culminating in visualization on Kentyou management platform.

Acknowledgment

This work was supported in part by NSF grants CNS-1652115, CNS-1827923, CNS-1910757, OAC-2029295, CNS-2148128, EEC-2133516, NSF grant CNS-2038984 and corresponding support from the Federal Highway Administration (FHA), ARO grants W911NF2210031 and W911NF1910379, and European Unions' H2020 R&D program through the NGI Explorers and NGI Atlantic projects under grant agreements No 825183 and No 871582, respectively. We thank Ivan Seskar, Jennifer Shane (Rutgers/WINLAB), and Owen Nuckolls (Columbia) for their support.

References

[1] D. Raychaudhuri, I. Seskar, G. Zussman, T. Korakis, D. Kilper, T. Chen, J. Kolodziejski, M. Sherman, Z. Kostic, X. Gu, H. Krishnaswamy, S. Maheshwari, P. Skrimponis, and C. Gutterman, "Challenge: COSMOS: A city-scale programmable testbed for experimentation with advanced wireless," in *Proc. ACM MobiCom'20*, 2020.

[2] Z. Kostic, A. Angus, Z. Yang, Z. Duan, I. Seskar, G. Zussman, and D. Raychaudhuri, "Smart city intersections: Intelligence nodes for future metropolises," *IEEE Comp.*, vol. 55, no. 12, pp. 74–85, 2022.

[3] COSMOS Project, "Hardware: Cameras," 2022. [Online]. Available: https://wiki.cosmos-lab.org/wiki/Hardware/Cameras.

[4] R. A. Light, "Mosquitto: Server and client implementation of the MQTT protocol," *J. Open Source Software*, vol. 2, no. 13, p. 265, 2017.

[5] L. Gürgen, C. Munilla, R. Druilhe, E. Gandrille, and J. Botelho do Nascimento, "Sensinact IoT platform as a service," *Enablers for Smart Cities*, pp. 127–147, 2016.

[6] M. Ghasemi, S. Kleisarchaki, T. Calmant, L. Gürgen, J. Ghaderi, Z. Kostic, and G. Zussman, "Real-time camera analytics for enhancing traffic intersection safety," in *Proc. ACM MobiSys'22*, 2022.

[7] A. Afzal, H. U. Draz, M. Z. Khan, and M. U. G. Khan, "Automatic helmet violation detection of motorcyclists from surveillance videos using deep learning approaches of computer vision," in *Proc. ICAI'21*, 2021.

[8] K.-T. Nguyen, D.-T. Dinh, M. N. Do, and M.-T. Tran, "Anomaly detection in traffic surveillance videos with GAN-based future frame prediction," in *Proc. Int. Conf. Multimedia Retr.*, 2020.

[9] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, "Real-time video analytics: The killer app for edge computing," *IEEE Comp.*, vol. 50, no. 10, pp. 58–67, 2017.

[10] X. Zhang, Y. Feng, P. Angeloudis, and Y. Demiris, "Monocular visual traffic surveillance: A review," *IEEE Tran. Intell. Transp. Sys.*, 2022.

[11] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[12] M. Ghasemi, Z. Kostic, J. Ghaderi, and G. Zussman, "Auto-SDA: Automated video-based social distancing analyzer," in *Proc. ACM HotEdgeVideo'21*, 2021.

[13] Z. Yang, M. Sun, H. Ye, Z. Xiong, G. Zussman, and Z. Kostic, "Bird's-eye view social distancing analysis system," in *ICC Workshops'22*, 2022.

[14] G. Bradski, "The OpenCV Library," *Dr. Dobb's J. Software Tools*, 2000.